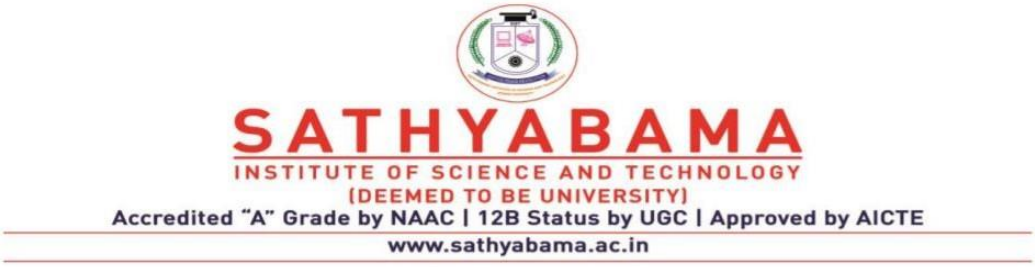

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT – II – SBS1203 – COMPUTER ARCHITECTURE

ADDER
- Half Adder
- Full Adder

SUBTRACTOR
- Half Subtractor
- Full Subractor

DECODER
ENCODER
MULTIPLEXER
DEMULTIPLEXER

2.1 ADDER

Adder is digital Logic Circuit in digital Electronics , used for addition of numbers.
In many computers and components like Processors, adders are not only used in arithmetic logic unit, and other operations like calculate the addresses, increment and decrement operators.
Two types: 1. Half Adder, 2. Full Adder
Have a look of single digit Addition,
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (with carry 1)

2.1.1. HALF ADDER

Half Adder adds two single digits. Digit A and Digit B. It has two output Sum(S) and carry (C)
The carry signal represents an overflow into the next digit of multi- digit Addition. Fig2.1 Shows the logical diagram of the half adder, which is the combination of the OR and AND gates. The OR gate output considered as Sum and the AND gat output considered as carry. Fig.2.2 shows the truth table of the half adder.

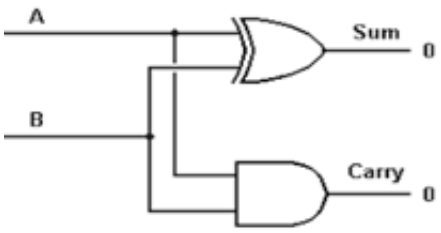

Fig 2.1 . Logical Circuit of the half adder

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fig.2.1.Truth Table

2.1.2. FULL ADDER

A Full Adder adds binary numbers and accounts for adding carried in as well as out. A one bit Full Adder has three inputs, often written as A,B and C_{in}.It has two outputs Sum (S) and Carry (C_{out}).The Main Difference between half adder and full adder is Full adder has three inputs and Two outputs.

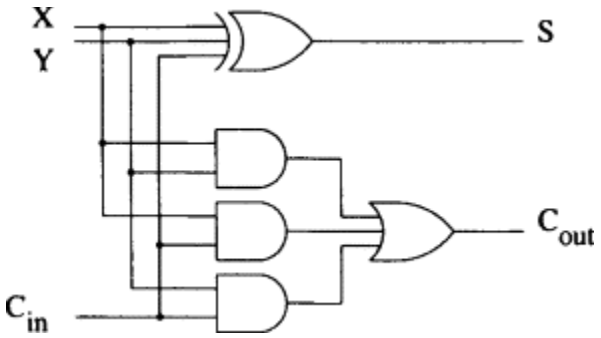

Fig.2.3. Logical Circuit of the Full adder

The Above diagram shows the full adder logical diagram, it is a combination of the two OR gates and three AND gates.

Inputs			Outputs	
A	B	C_{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig.2.4. Truth Table of the Full Adder

2.2. SUBTRACTOR

The Subtraction of the two number by taking the complement of the subtrahend and adding it to the minuend. By this method, the subtraction operation becomes an addition operation requiring full-adder for its machine implementation. Subtraction operation is each subtrahend bit of the number is subtracted from the minuend bit to form a difference bit. If the minuend bit is smaller than the subtrahend bit, 1 is borrowed from the next significant bit. The fact that a 1 has been borrowed from the next significant bit.

2.2.1. HALF SUBTRACTOR

Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow). It produces the difference between the two binary bits at the input and also produces an output (Borrow) to indicate if a 1 has been borrowed. In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.

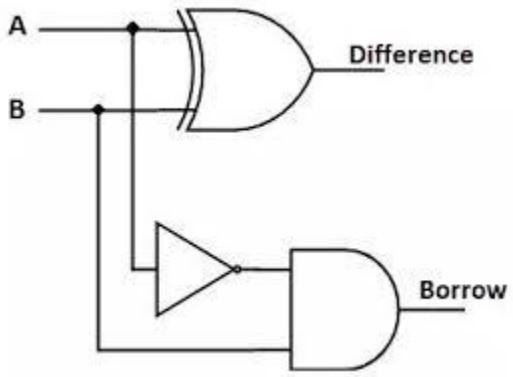

Fig.2.5. Half Subtractor Logical Circuit

Inputs		Outputs	
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Fig.2.6.Truth Table of Half Subtractor

2.2.2. FULL SUBTRACTOR

The full subtractor is a combinational circuit with three inputs A,B,C and two output D and C'. A is the 'minuend', B is 'subtrahend', C is the 'borrow' produced by the previous stage, D is the difference output and C' is the borrow output.

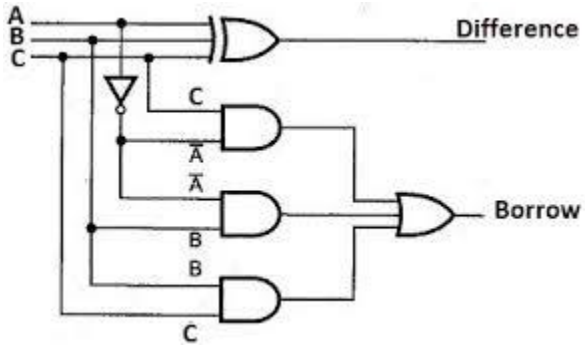

Fig.2.7. Logical Circuit of Full Subtractor

Inputs			Outputs	
A	B	$Borrow_{in}$	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Fig.2.8. Truth Table of Full Subtractor

2.3. ENCODER

Encoder is a combinational circuit which is designed to perform the inverse operation of the decoder. An encoder has n number of input lines and m number of output lines. An encoder produces an m bit binary code corresponding to the digital input number. The encoder accepts an n input digital word and converts it into an m bit another digital word.

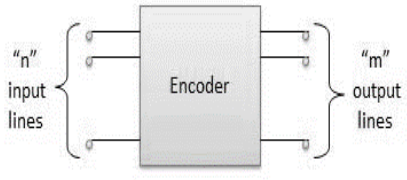

Fig.2.9. Block Diagram of the Encoder

Examples of Encoder

- Priority encoders
- Decimal to BCD encoder
- Octal to binary encoder

- Hexadecimal to binary encoder

2.3.1. PRIORITY ENCODER

This is a special type of encoder. Priority is given to the input lines. If two or more input line are 1 at the same time, then the input line with highest priority will be considered. There are four input D_0, D_1, D_2, D_3 and two output Y_0, Y_1. Out of the four input D_3 has the highest priority and D_0 has the lowest priority. if $D_3 = 1$ then $Y_1\ Y_1 = 11$ irrespective of the other inputs. Similarly if $D_3 = 0$ and $D_2 = 1$ then $Y_1\ Y_0 = 10$ irrespective of the other inputs.

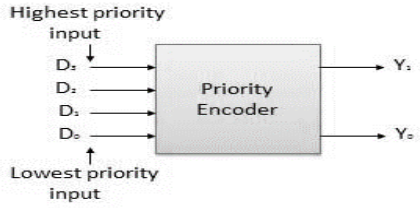

Fig.2.10. Block Diagram of the Priority Encoder

Highest	Inputs		Lowest	Outputs	
D_3	D_2	D_1	D_0	Y_0	Y_1
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

Fig.2.11. Truth Table of the Priority Encoder

2.4. MULTIPLEXER

Multiplexer is a special type of combinational circuit. There are n-data inputs, one output and m select inputs with $2^m = n$. It is a digital circuit which selects one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the selected inputs. Depending on the digital code applied at the selected inputs, one out of n data sources is selected and transmitted to the single output Y.

E is called the strobe or enable input which is useful for the cascading. It is generally an active low terminal that means it will perform the required operation when it is low. This is called as Data Selector.

Types of Multiplexer :

2 : 1 multiplexer

4 : 1 multiplexer

16 : 1 multiplexer

32 : 1 multiplexer

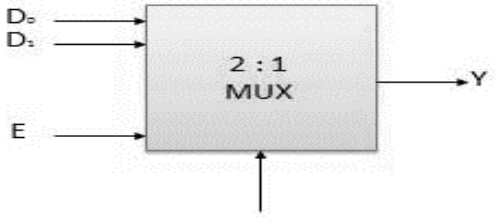

Fig. 2.12. Block diagram of the Multilexer

Enable	Select	Output
E	S	Y
0	x	0
1	0	D_0
1	1	D_1

x = Don't care

Fig.2.13. Truth table of the Multiplexer

2.5. DEMULPLEXER

A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. It has only one input, n outputs, m select input. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line.

Demultiplexers in multiple variations.

1: 2 demultiplexer
1 : 4 demultiplexer

1 : 16 demultiplexer
1 : 32 demultiplexer

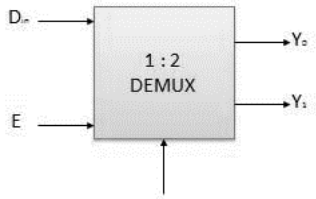

Fig. 2.14. Block Diagram of Demultiplexer

Enable	Select	Output
E	S	Y0 Y1
0	x	0 0
1	0	0 D_{in}
1	1	D_{in} 0

x = Don't care

Fig.2.15. Truth Table of the Demultiplexer

2.6. SEQUENTIAL LOGIC

The combinational circuit does not use any memory. Hence the previous state of input does not have any effect on the present state of the circuit. But sequential circuit has memory so output can vary based on input. This type of circuits uses previous input, output, clock and a memory element.

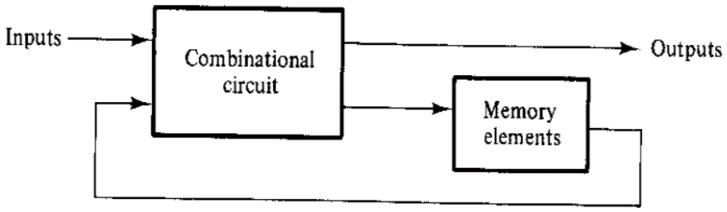

FIGURE 6-1
Block diagram of a sequential circuit

Fig.2.16. Block diagram of a Sequential Circuit

The following all are the sequential logic circuits,

- FLIPFLOP
- R-S FLIP FLOP
- M-S JK FLIPFLOP
- DELAY FLIPFLOP
- TOGGLE FLIPFLOP
- REGISTERS
- COUNTERS

2.7. FLIP FLOP

Flip flop is a sequential circuit which generally samples its inputs and changes its outputs only at particular instants of time and not continuously. Flip flop is said to be edge sensitive or edge triggered rather than being level triggered like latches.

2.7.1. S-R FLIP FLOP

It is basically S-R latch using NAND gates with an additional enable input. It is also called as level triggered SR-FF. For this, circuit in output will take place if and only if the enable input (E) is made active. In short this circuit will operate as an S-R latch if E = 1 but there is no change in the output if E = 0.

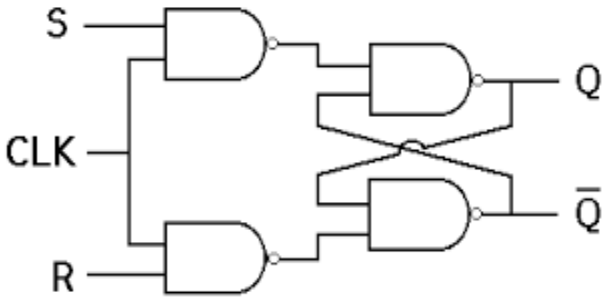

Fig. 2.17. Logical Diagram of S-R FlipFlop

Inputs			Outputs		Comments
E	S	R	Q_{n+1}	$\overline{Q_{n+1}}$	
1	0	0	Q_n	$\overline{Q_n}$	No change
1	0	1	0	1	Rset
1	1	0	1	0	Set
1	1	1	x	x	Indeterminate

Fig.2.18. Truth Table of S-R FlipFlop

S-R FLIP FLOP – OPERATIONS

S = R = 0 : No change

If S = R = 0 then output of NAND gates 3 and 4 are forced to become 1.

Hence R' and S' both will be equal to 1. Since S' and R' are the input of the basic S-R latch using NAND gates, there will be no change in the state of outputs.

S = 0, R = 1, E = 1

Since S = 0, output of NAND-3 i.e. R' = 1 and E = 1 the output of NAND-4 i.e. S' = 0.

Hence Qn+1 = 0 and Qn+1 bar = 1. This is reset condition.

S =0 1, R = 0, E = 1

Output of NAND-3 i.e. R' = 0 and output of NAND-4 i.e. S' = 1.

Hence output of S-R NAND latch is Qn+1 = 1 and Qn+1 bar = 0. This is the reset condition.

- **S = 1, R = 1, E = 1**

  As S = 1, R = 1 and E = 1, the output of NAND gates 3 and 4 both are 0 i.e. S' = R' = 0. Hence the Race condition will occur in the basic NAND latch.

7.2. MASTER SLAVE JK FLIPFLOP

Master slave JK FF is a cascade of two S-R FF with feedback from the output of second to input of first. Master is a positive level triggered. But due to the presence of the inverter in the clock line, the slave will respond to the negative level. Hence when the clock = 1 (positive level) the master is active and the slave is inactive. Whereas when clock = 0 (low level) the slave is active and master is inactive.

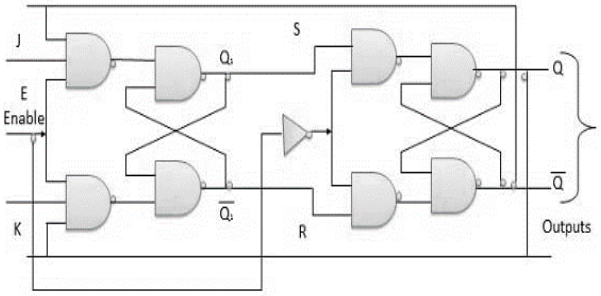

Fig.2.19.Logical Diagram of M-S FlipFlop

Inputs			Outputs		Comments
E	J	K	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	0	Q_n	$\overline{Q}_n$	No change
1	0	1	0	1	Rset
1	1	0	1	0	Set
1	1	1	$\overline{Q}_n$	Q_n	Toggle

Fig.2.20. Truth Table of M-S FlipFlop

MS JK FLIP FLOP –OPERATIONS

- **J = K = 0 (No change)**

  When clock = 0, the slave becomes active and master is inactive. But since the S and R inputs have not changed, the slave outputs will also remain unchanged. Therefore outputs will not change if J = K =0.

- **J = 0 and K = 1 (Reset)**
  Clock = 1 − Master active, slave inactive. Therefore outputs of the master become $Q_1 = 0$ and Q_1 bar = 1. That means S = 0 and R =1.Clock = 0 − Slave active, master inactive. Therefore outputs of the slave become Q = 0 and Q bar = 1.
  Again clock = 1 − Master active, slave inactive. Therefore even with the changed outputs Q = 0 and Q bar = 1 fed back to master, its output will be Q1 = 0 and Q1 bar = 1. That means S = 0 and R = 1. Hence with clock = 0 and slave becoming active the outputs of slave will remain Q = 0 and Q bar = 1. Thus we get a stable output from the Master slave.
- **J = 1 and K = 0 (Set)**
  Clock = 1 − Master active, slave inactive. Therefore outputs of the master become $Q_1 = 1$ and Q_1 bar = 0. That means S = 1 and R =0. Clock = 0 − Slave active, master inactive. Therefore outputs of the slave become Q = 1 and Q bar = 0. Again clock = 1 − then it can be shown that the outputs of the slave are stabilized to Q = 1 and Q bar = 0.
- **J = K = 1 (Toggle)**
  Clock = 1 − Master active, slave inactive. Outputs of master will toggle. So S and R also will be inverted. Clock = 0 − Slave active, master inactive. Outputs of slave will toggle. These changed output are returned back to the master inputs. But since clock = 0, the master is still inactive. So it does not respond to these changed outputs. This avoids the multiple toggling which leads to the race around condition. The master slave flip flop will avoid the race around condition.

2.7.3. DELAY FLIP FLOP / D FLIPFLOP

Delay Flip Flop or D Flip Flop is the simple gated S-R latch with a NAND inverter connected between S and R inputs. It has only one input. The input data is appearing at the output after some time.

Due to this data delay between i/p and o/p, it is called delay flip flop. S and R will be the complements of each other due to NAND inverter. Hence S = R = 0 or S = R = 1, these input condition will never appear. This problem is avoid by SR = 00 and SR = 1 conditions.

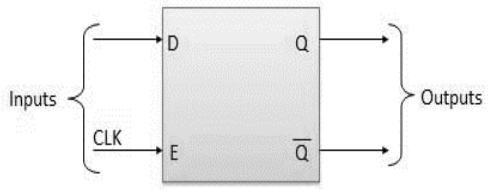

Fig.2.21. Block Diagram of D- FlipFlop

Inputs		Outputs		Comments
E	D	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	0	1	Rset
1	1	1	0	Set

Fig.2.22. Truth Table of the D- FlipFlop

D FLIPFLOP – Operations

E = 0

Latch is disabled. Hence no change in output.

E = 1 and D = 0

If E = 1 and D = 0 then S = 0 and R = 1. Hence irrespective of the present state, the next state is Qn+1 = 0 and Qn+1 bar = 1. This is the reset condition.

E = 1 and D = 1

If E = 1 and D = 1, then S = 1 and R = 0. This will set the latch and Q_{n+1} = 1 and Q_{n+1} bar = 0 irrespective of the present state.

7.4. TOGGLE FLIP FLOP / T FLIP FLOP

Toggle flip flop is basically a JK flip flop with J and K terminals permanently connected together. It has only input denoted by T as shown in the Symbol Diagram. The symbol for positive edge triggered T flip flop is shown in the Block Diagram.

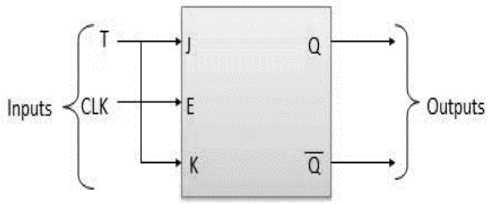

Fig.2.23. Block Diagram of T- FlipFlop

Inputs		Outputs		Comments
E	T	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	Q_n	$\overline{Q}_n$	No change
1	1	$\overline{Q}_n$	Q_n	Toggle

Fig.2.24.Truth Table of T-FlipFlop

T FLIP FLOP – Operations

T = 0, J = K = 0

The output Q and Q bar won't change

T = 1, J = K = 1

Output will toggle corresponding to every leading edge of clock signal.

2.8. DIGITAL REGISTERS

Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a **Register**.

The **n-bit register** will consist of **n** number of flip-flop and it is capable of storing an **n-bit** word.

The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.

- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

2.8.1. Serial Input Serial Output

Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to D_{in} bit with the LSB bit applied first. The D input of FF-3 i.e. D_3 is connected to serial data input D_{in}. Output of FF-3 i.e. Q_3 is connected to the input of the next flip-flop i.e. D_2 and so on.

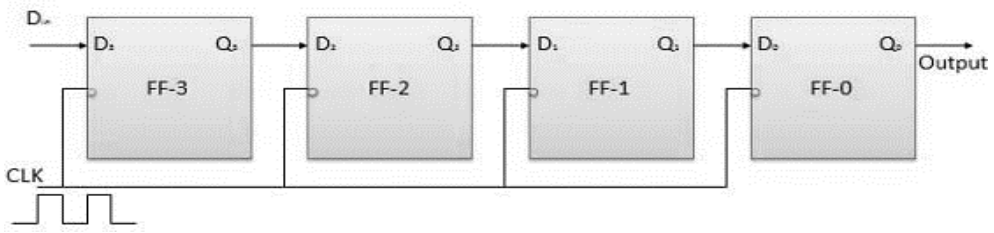

Fig.2.24. Block Diagram of the Serial In Serial Out

Operation: Before application of clock signal, let Q3 Q2 Q1 Q0 = 0000 and apply LSB bit of the number to be entered to Din. So Din = D3 = 1. Apply the clock. On the first falling edge of clock, the FF-3 is set, and stored word in the register is Q3 Q2 Q1 Q0 = 1000.

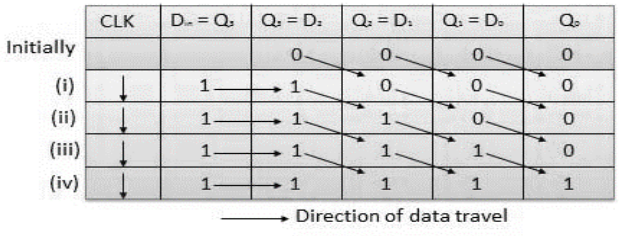

Fig.2.25. Truth Table of SISO

2.8.2.Serial Input Parallel Output

n such types of operations, the data is entered serially and taken out in parallel fashion.Data is loaded bit by bit. The outputs are disabled as long as the data is loading.As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.

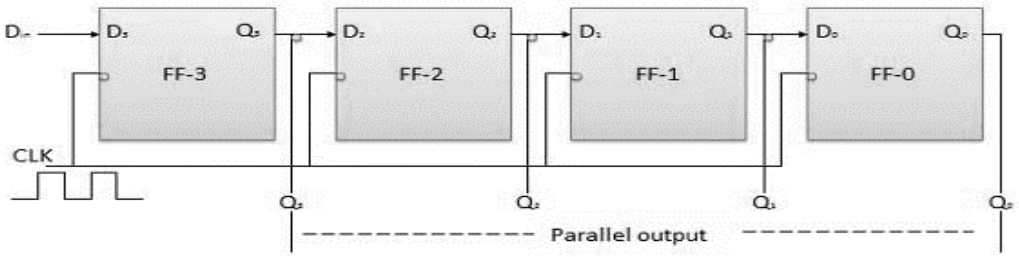

Fig.2.25. Block Diagram of SIPO

2.8.3.Parallel Input Serial Output (PISO)

Data bits are entered in parallel fashion. The circuit shown below is a four bit parallel input serial output register. Output of previous Flip Flop is connected to the input of the next one via a combinational circuit. The binary input word B_0, B_1, B_2, B_3 is applied though the same combinational circuit. There are two modes in which this circuit can work namely - shift mode or load mode.

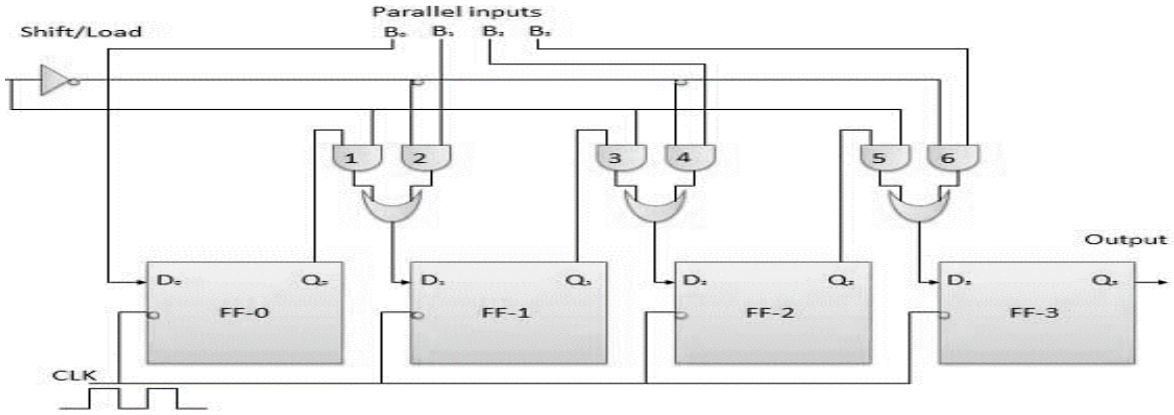

Fig.2.26. Parallel Input Serial Output

Load Mode

When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

When the shift/load bar line is low (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

2.8.4.Parallel Input Parallel Output (PISO)

The 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.

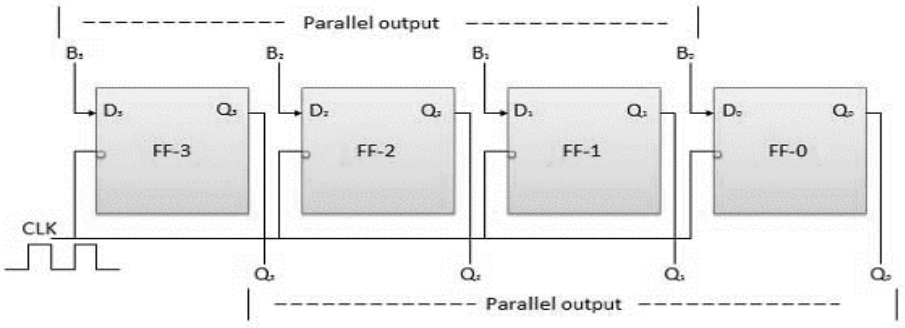

Fig.27. Block Diagram of Parallel Input Parallel Output

2.9.BIDIRECTIONAL SHIFT REGISTER

If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig2.28. There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

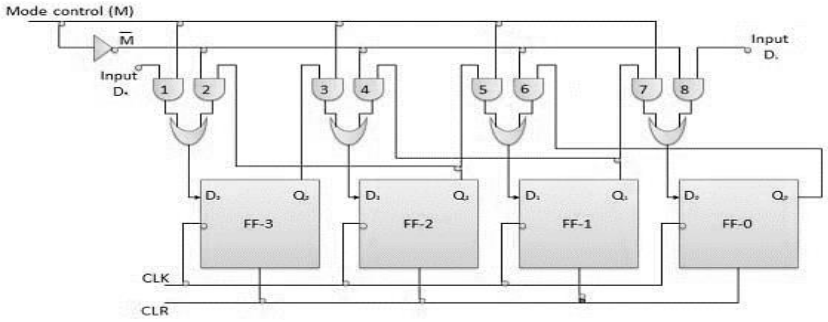

Fig.2.28. Block Diagram of bidirectional shift register

Operations :

With M = 1 − Shift right operation

If M = 1, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled.

The data at D_R is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with M = 1 we get the serial right shift operation.

With M = 0 − Shift left operation

When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled.

The data at D_L is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with M = 0 we get the serial right shift operation.

2.10. UNIVERSAL SHIFT REGISTER

A shift register which can shift the data in only one direction is called a unidirectional shift register. A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can

shift the data in both directions as well as load it parallely, is known as a universal shift register.

The shift register is capable of performing the following operation −

- Parallel loading
- Left Shifting
- Right shifting

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected to ground, the universal shift register acts as a bi-directional register.

For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.

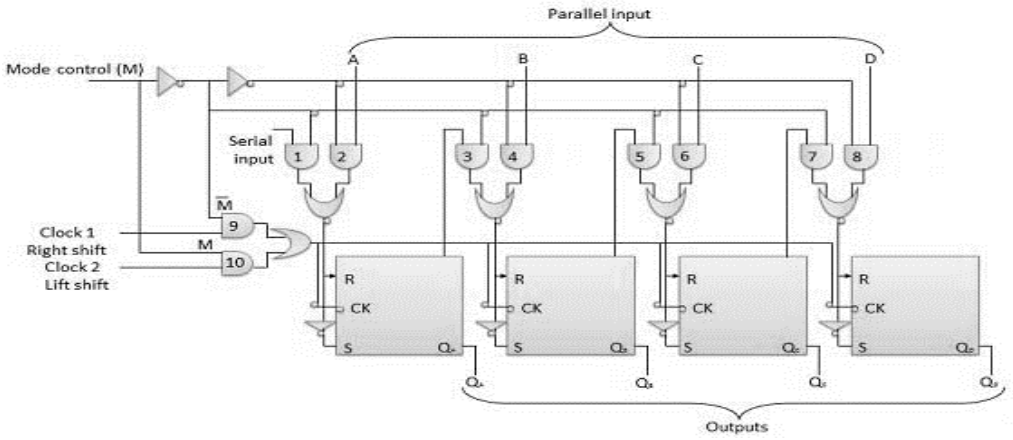

Fig.2.29. Block Diagram UNIVERSAL SHIFT REGISTER

2.11.COUNTERS

Counter is a sequential circuit. A digital circuit which is used for a counting pulses is known counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied.

Counters are of two types:

- Asynchronous or ripple counters.
- Synchronous counters.

2.11.1.ASYNCHRONOUS OR RIPPLE COUNTERS

The logic diagram of a 2-bit ripple up counter is shown in figure. The toggle (T) flip-flop are being used. But we can use the JK flip-flop also with J and K connected permanently to logic 1. External clock is applied to the clock input of flip-flop A and Q_A output is applied to the clock input of the next flip-flop i.e. FF-B.

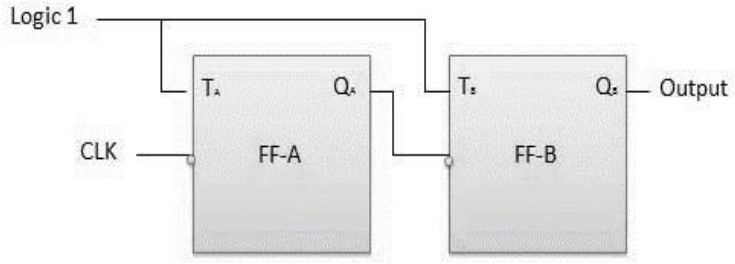

Fig.2.30. Block Diagram of Asynchronous Or Ripple Counters

Operations :

Initially let both the FFs be in the reset state

$Q_BQ_A = 00$ initially

After 1st negative clock edge.

As soon as the first negative clock edge is applied, FF-A will toggle and Q_A will be equal to 1.Q_A is connected to clock input of FF-B. Since Q_A has changed from 0 to 1, it is treated as the positive clock edge by FF-B. There is no change in Q_B because FF-B is a negative edge triggered FF. $Q_BQ_A = 01$ after the first clock pulse.

After 2nd negative clock edge

On the arrival of second negative clock edge, FF-A toggles again and $Q_A = 0$.

The change in Q_A acts as a negative clock edge for FF-B. So it will also toggle, and Q_B will be 1. $Q_BQ_A = 10$ after the second clock pulse.

After 3rd negative clock edge

On the arrival of 3rd negative clock edge, FF-A toggles again and Q_A become 1 from 0.

Since this is a positive going change, FF-B does not respond to it and remains inactive.

So Q_B does not change and continues to be equal to 1.

$Q_BQ_A = 11$ after the third clock pulse.

After 4th negative clock edge

On the arrival of 4th negative clock edge, FF-A toggles again and Q_A becomes 1 from 0.

This negative change in Q_A acts as clock pulse for FF-B. Hence it toggles to change Q_B from 1 to 0. $Q_BQ_A = 00$ after the fourth clock pulse.

Clock	Counter output		State number	Deciimal Counter output
	Q_B	Q_A		
Initially	0	0	—	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0

Fig.2.30. Truth Table of the Ripple Counter

2.11.2. SYNCHRONOUS COUNTERS

If the "clock" pulses are applied to all the flip-flops in a counter simultaneously, then such a counter is called as synchronous counter.

2-bit Synchronous up counter

The J_A and K_A inputs of FF-A are tied to logic 1. So FF-A will work as a toggle flip-flop. The J_B and K_B inputs are connected to Q_A.

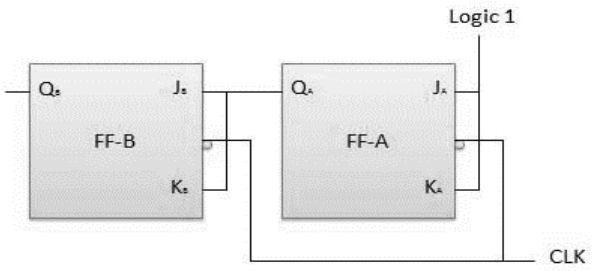

Fig.2.31. Block Diagram of Synchronous Counters

Operations:

Initially let both the FFs be in the reset state

Q_BQ_A = 00 initially.

After 1st negative clock edge

As soon as the first negative clock edge is applied, FF-A will toggle and Q_A will change from 0 to 1.

But at the instant of application of negative clock edge, Q_A , $J_B = K_B = 0$. Hence FF-B will not change its state. So Q_B will remain 0.

Q_BQ_A = 01 after the first clock puls.

After 2nd negative clock edge

On the arrival of second negative clock edge, FF-A toggles again and Q_A changes from 1 to 0.But at this instant Q_A was 1. So $J_B = K_B = 1$ and FF-B will toggle. Hence Q_B changes from 0 to 1.

Q_BQ_A = 10 after the second clock pulse.

After 3rd negative clock edge

On application of the third falling clock edge, FF-A will toggle from 0 to 1 but there is no change of state for FF-B.

Q_BQ_A = 11 after the third clock pulse.

After 4th negative clock edge

On application of the next clock pulse, Q_A will change from 1 to 0 as Q_B will also change from 1 to 0.

Q_BQ_A = 00 after the fourth clock pulse.

2.12.CLASSIFICATION OF COUNTERS

Depending on the way in which the counting progresses, the synchronous or asynchronous counters are classified as follows :

- Up counters
- Down counters
- Up/Down counters

Up counter and down counter is combined together to obtain an UP/DOWN counter. A mode control (M) input is also provided to select either up or down mode. A combinational circuit is required to be designed and used between each pair of flip-flop in order to achieve the up/down operation.

Type of up/down counters:

- UP/DOWN ripple counters
- UP/DOWN synchronous counter

2.12.1.UP/DOWN Ripple Counter

In the UP/DOWN ripple counter all the FFs operate in the toggle mode. So either T flip-flops or JK flip-flops are to be used. The LSB flip-flop receives clock directly. But the clock to every other FF is obtained from (Q = Q bar) output of the previous FF.

UP counting mode (M=0) − The Q output of the preceding FF is connected to the clock of the next stage if up counting is to be achieved. For this mode, the mode select input M is at logic 0 (M=0).

DOWN counting mode (M=1) − If M = 1, then the Q bar output of the preceding FF is connected to the next FF. This will operate the counter in the counting mode.

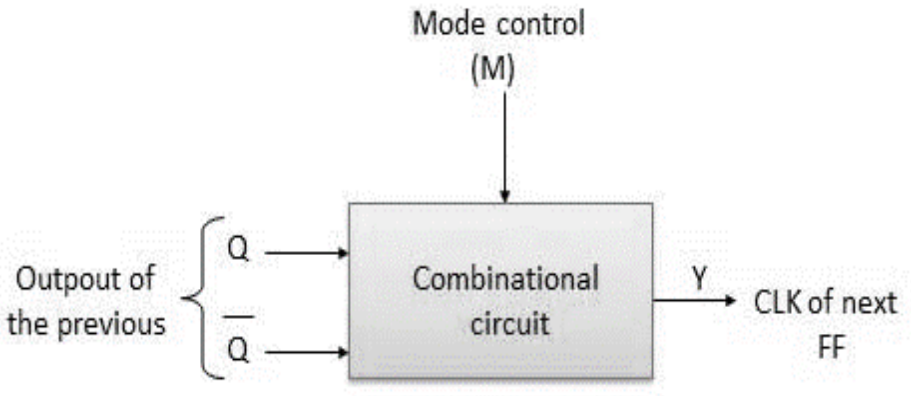

Fig.2.31. Combinational Circuit of UP/Down Ripple Counter

Inputs			Outputs	
M	Q	$\overline{Q}$	Y	
0	0	0	0	Y = Q for up counter
0	0	1	0	
0	1	0	1	
0	1	1	1	
1	0	0	0	$Y = \overline{Q}$ for up counter
1	0	1	1	
1	1	0	0	
1	1	1	1	

Fig.2.32. Truth Table of Ripple Counter

Operations:

Case 1 − With M = 0 (Up counting mode)

If M = 0 and M bar = 1, then the AND gates 1 and 3 in fig. will be enabled whereas the AND gates 2 and 4 will be disabled.

Hence Q_A gets connected to the clock input of FF-B and Q_B gets connected to the clock input of FF-C.

These connections are same as those for the normal up counter. Thus with M = 0 the circuit work as an up counter.

Case 2: With M = 1 (Down counting mode)

If M = 1, then AND gates 2 and 4 in fig. are enabled whereas the AND gates 1 and 3 are disabled.

Hence Q_A bar gets connected to the clock input of FF-B and Q_B bar gets connected to the clock input of FF-C.

These connections will produce a down counter. Thus with M = 1 the circuit works as a down counter.

The 2-bit ripple counter is called as MOD-4 counter and 3-bit ripple counter is called as MOD-8 counter. So in general, an n-bit ripple counter is called as modulo-N counter. Where, MOD number = 2^n.

Type of modulus

- 2-bit up or down (MOD-4)
- 3-bit up or down (MOD-8)
- 4-bit up or down (MOD-16)

Application of counters

- Frequency counters
- Digital clock
- Time measurement
- A to D converter
- Frequency divider circuits
- Digital triangular wave generator.